



# Reconocimiento de Patrones

Version 2022-2

**SVM**

**Dr. José Ramón Iglesias**

DSP-ASIC BUILDER GROUP

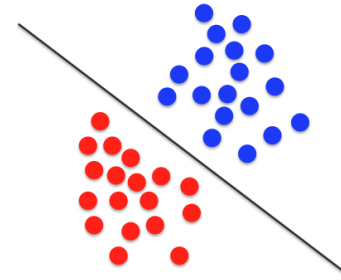
Director Semillero TRIAC

Ingeniería Electrónica

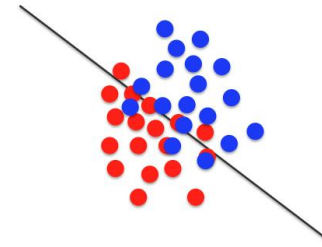
Universidad Popular del Cesar

# SVM: Máquinas vectoriales de soporte

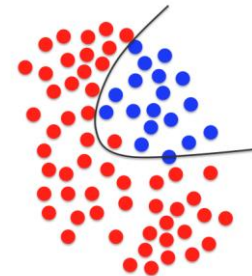
1) Lineal con separación perfecta



2) Lineal sin separación perfecta

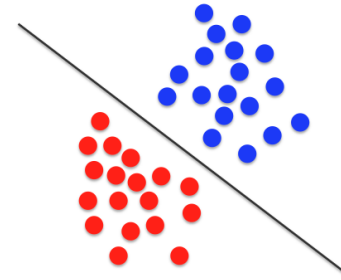


3) No lineal

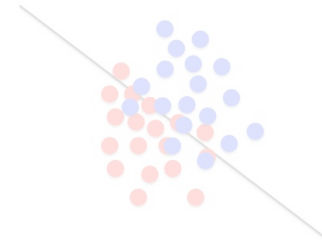


# SVM: Máquinas vectoriales de soporte

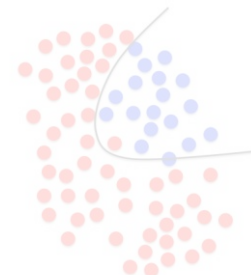
1) Lineal con separación perfecta



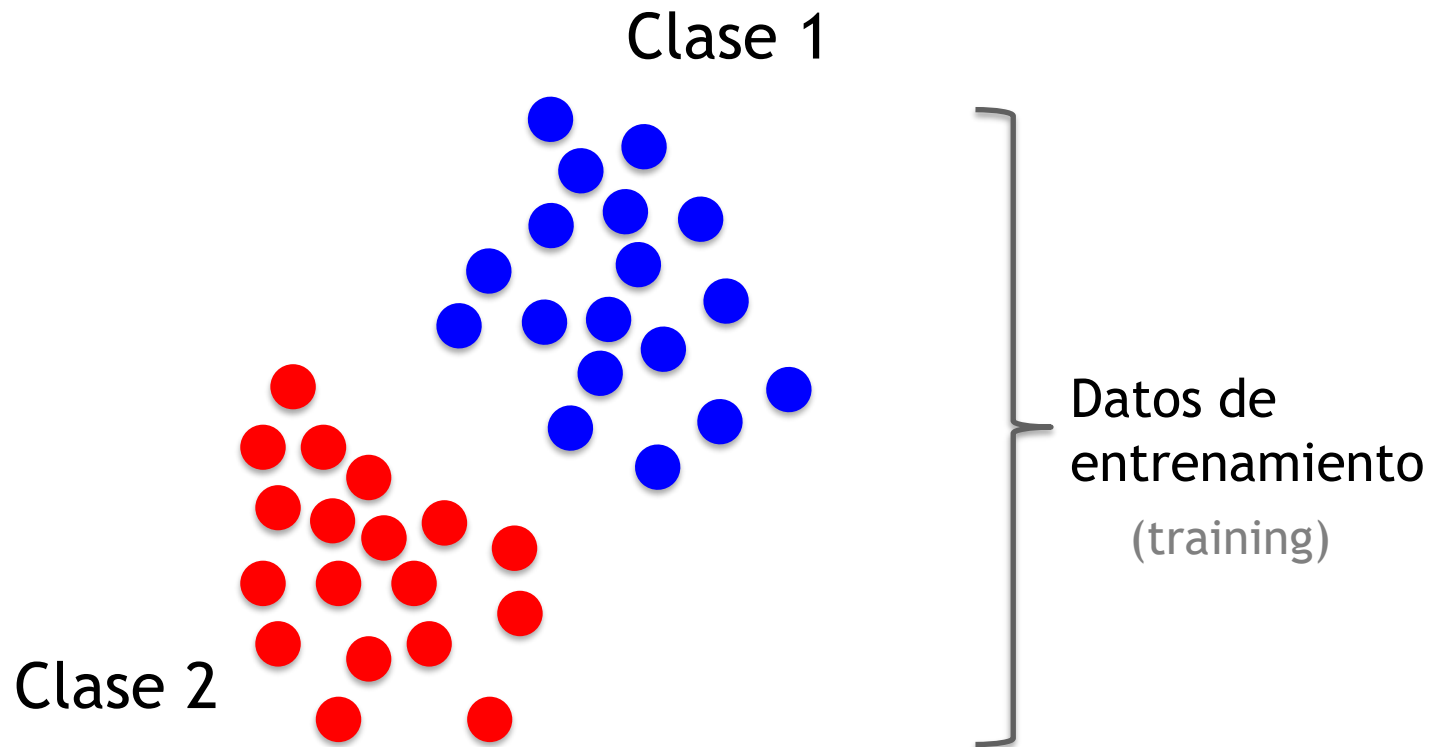
2) Lineal sin separación perfecta



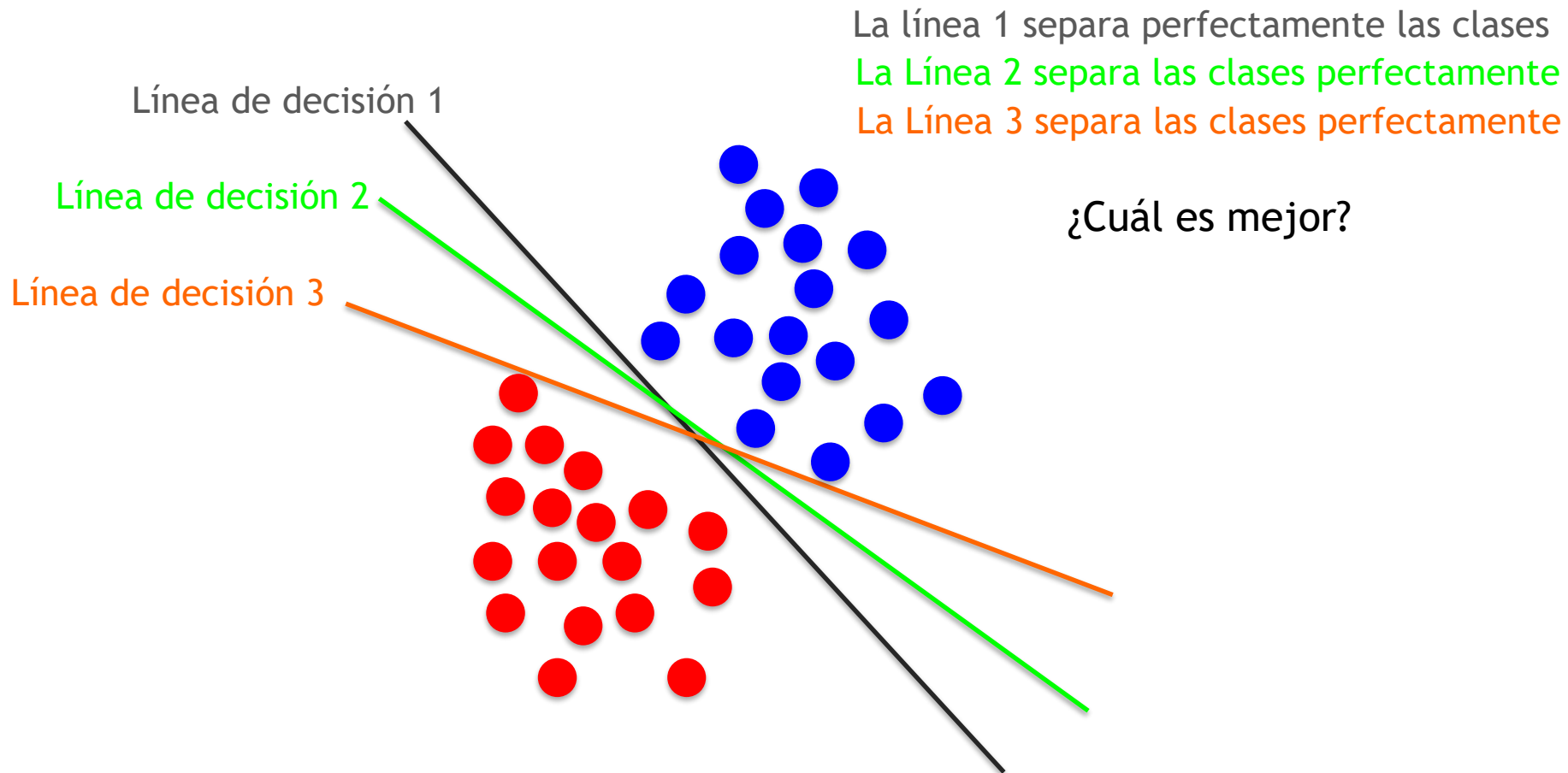
3) No lineal



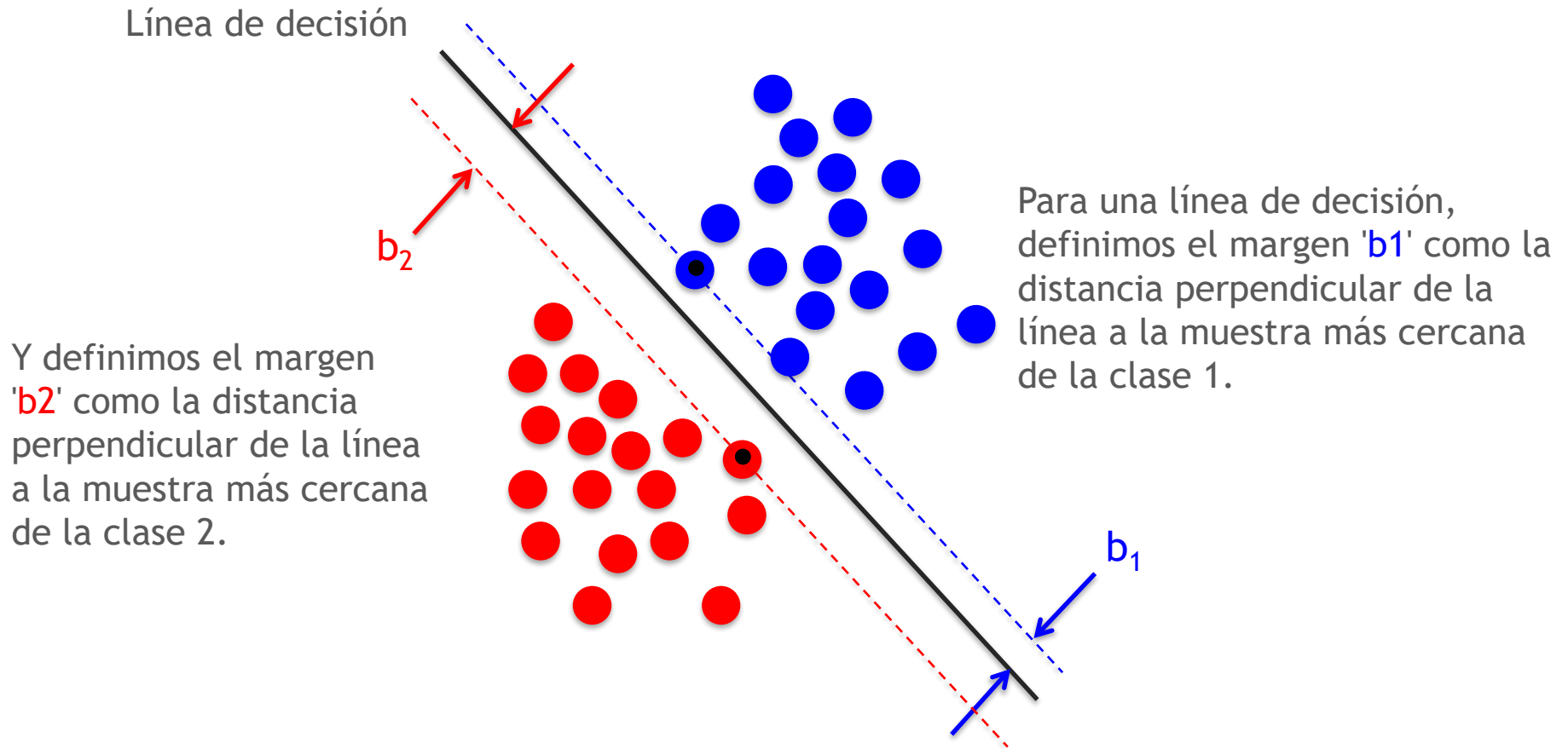
## SVM: dos clases



## SVM: dos clases



## SVM: dos clases

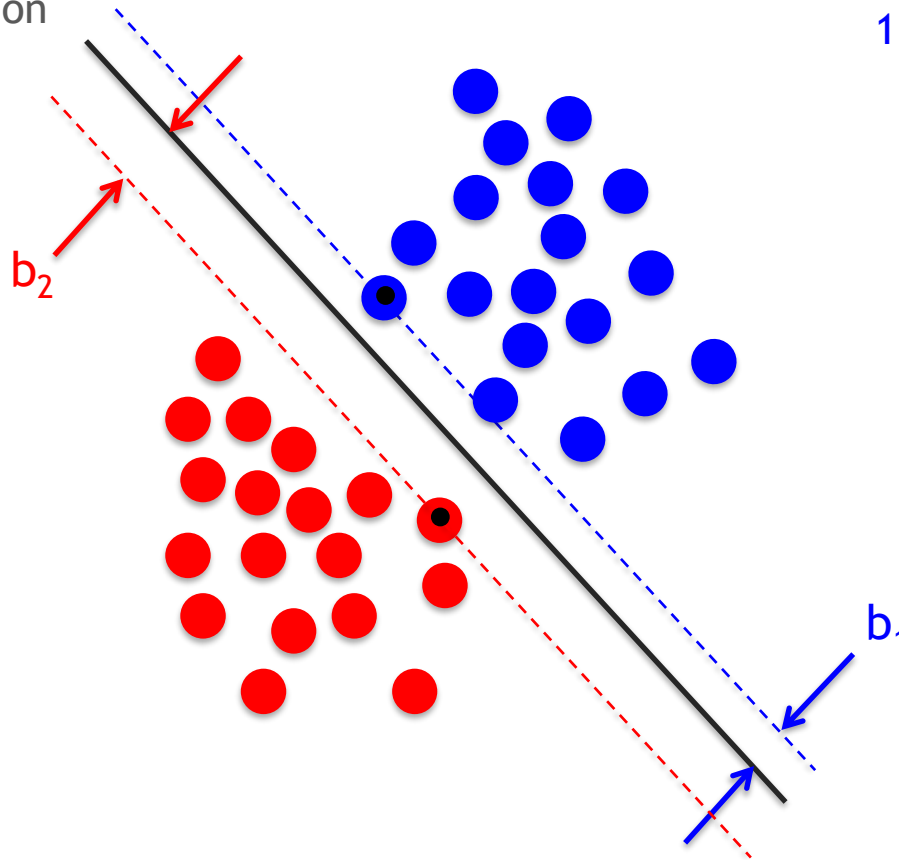


## SVM: dos clases

Ideas clave de SVM:

1)  $b_1 = b_2 = b$ .

Línea de decisión

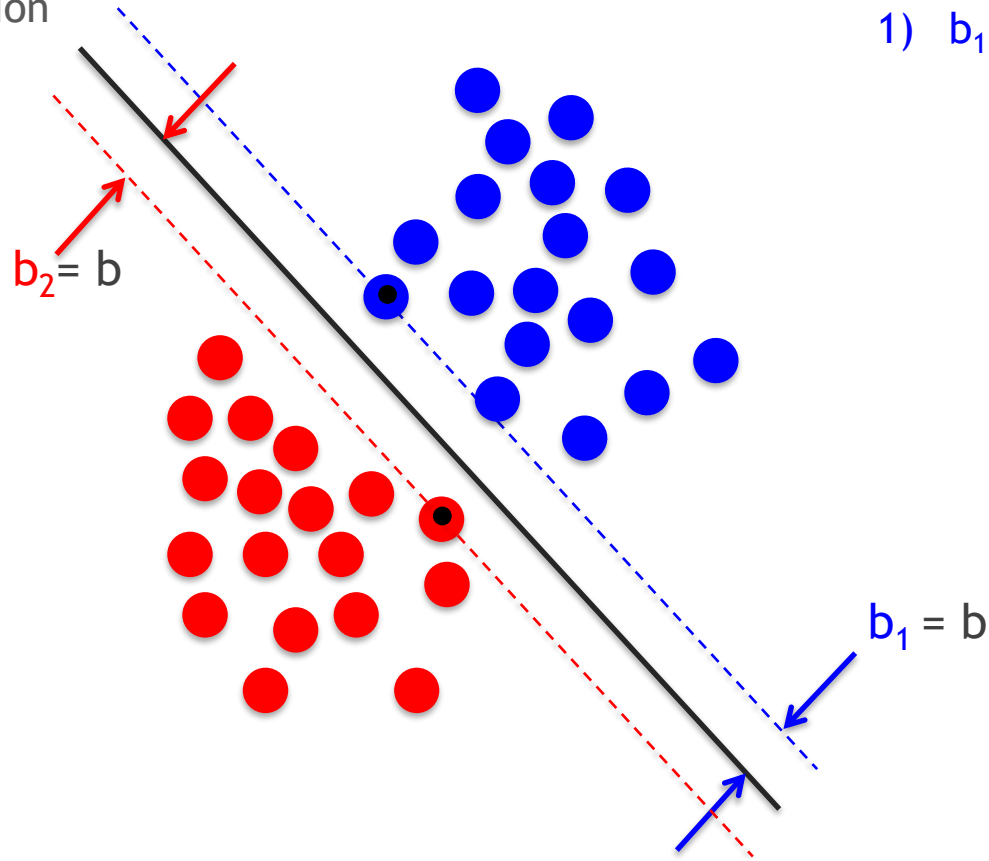


## SVM: dos clases

Ideas clave de SVM:

1)  $b_1 = b_2 = b$ .

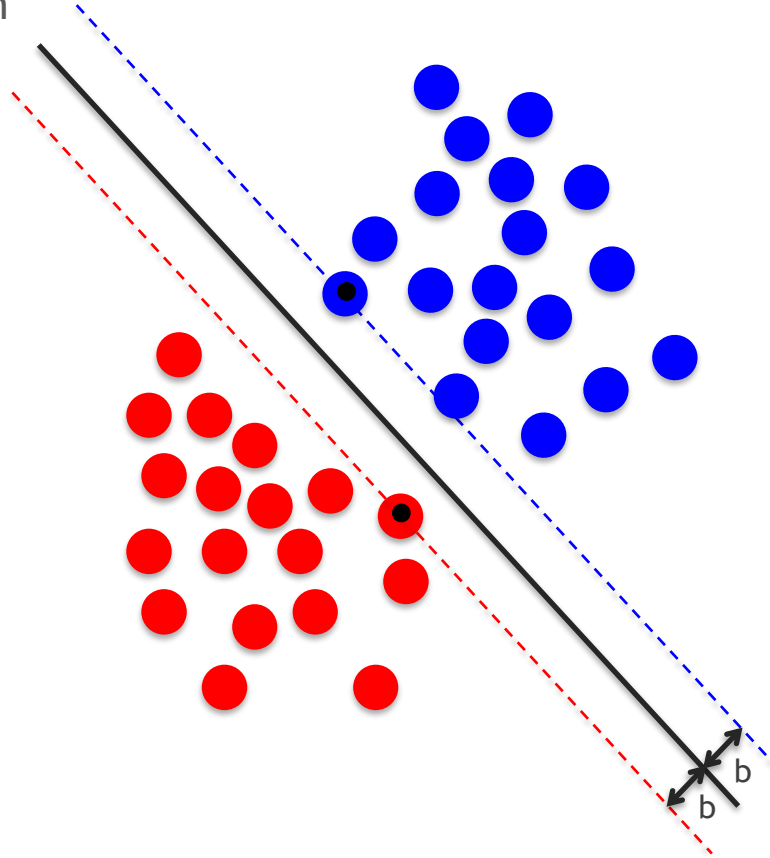
Línea de decisión





## SVM: dos clases

Línea de decisión

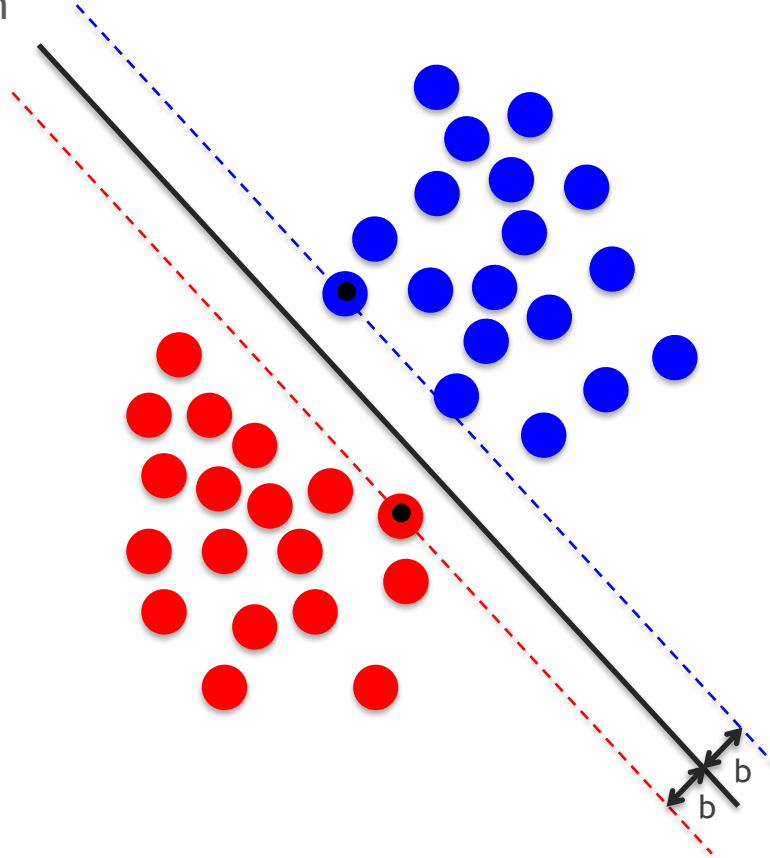


Ideas clave de SVM:

1)  $b_1 = b_2 = b$ .

## SVM: dos clases

Línea de decisión



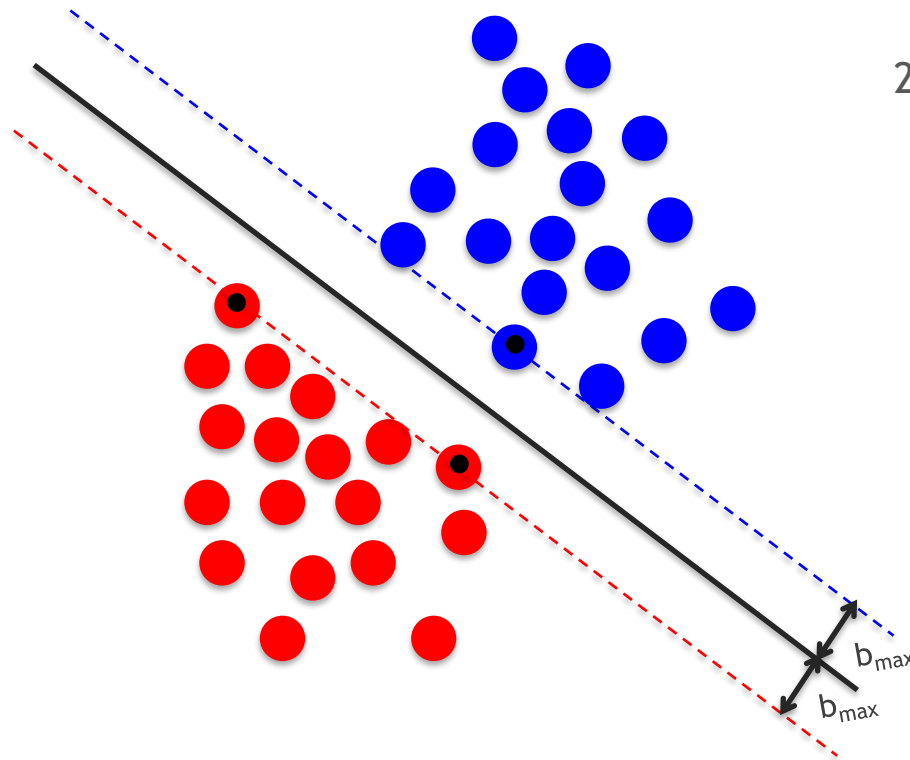
Ideas clave de SVM:

1)  $b_1 = b_2 = b$ .

2)  $b$  debe maximizarse.

## SVM: dos clases

Línea de decisión



Ideas clave de SVM:

1)  $b_1 = b_2 = b$ .

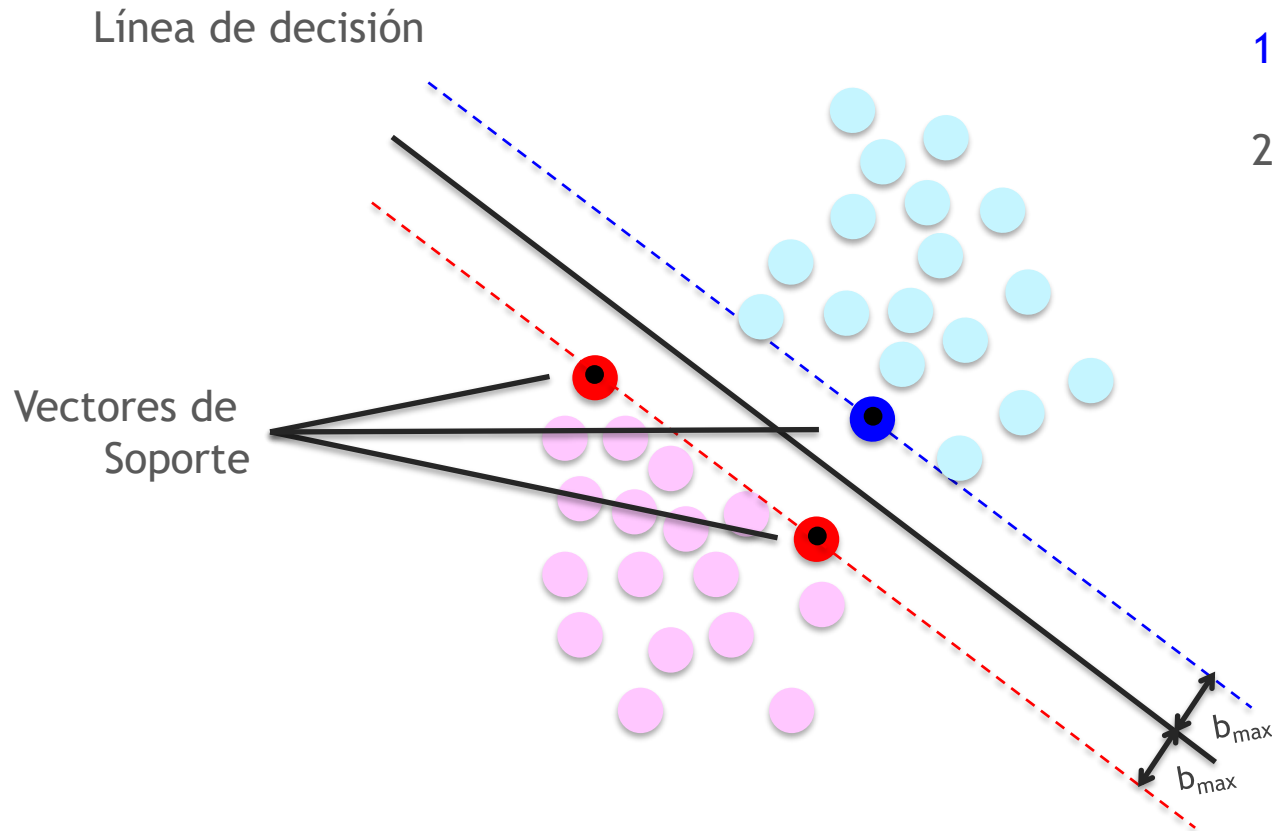
2)  $b$  debe maximizarse.

## SVM: dos clases

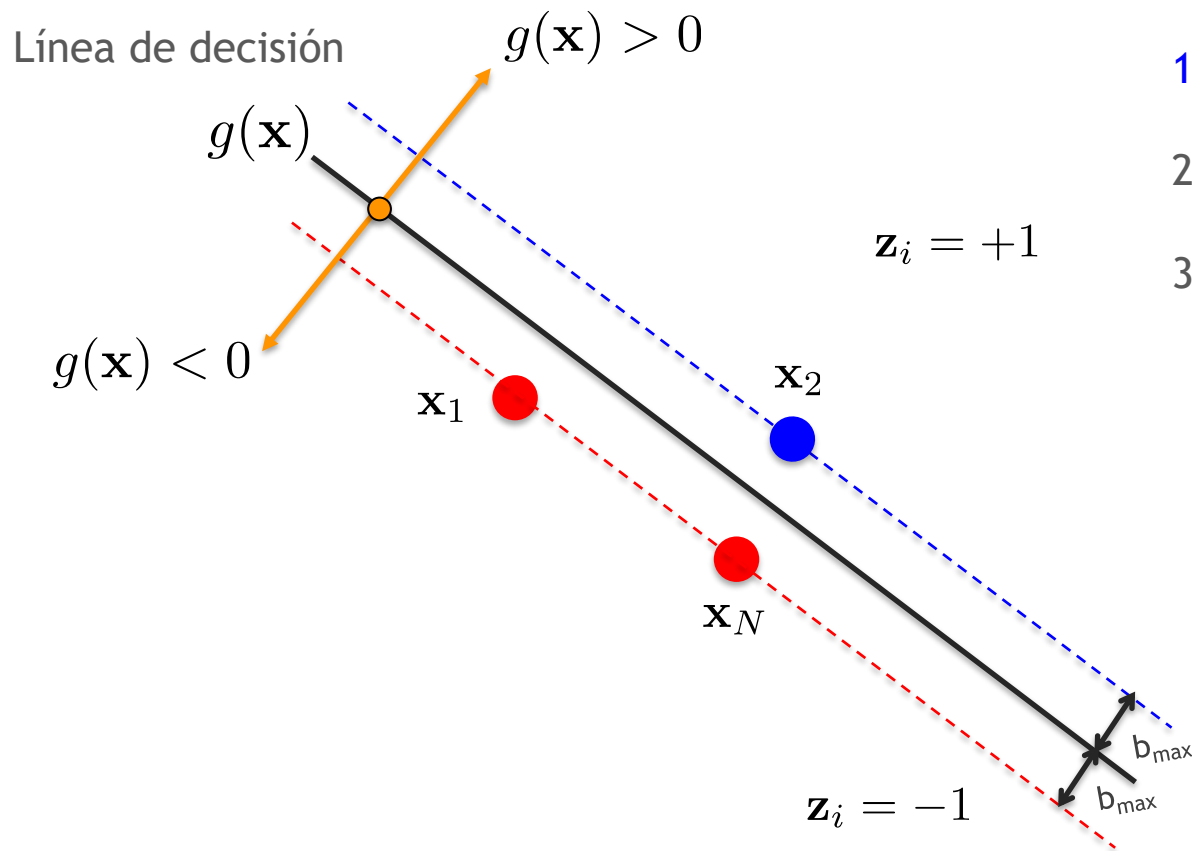
Ideas clave de SVM:

1)  $b_1 = b_2 = b$ .

2)  $b$  debe maximizarse.



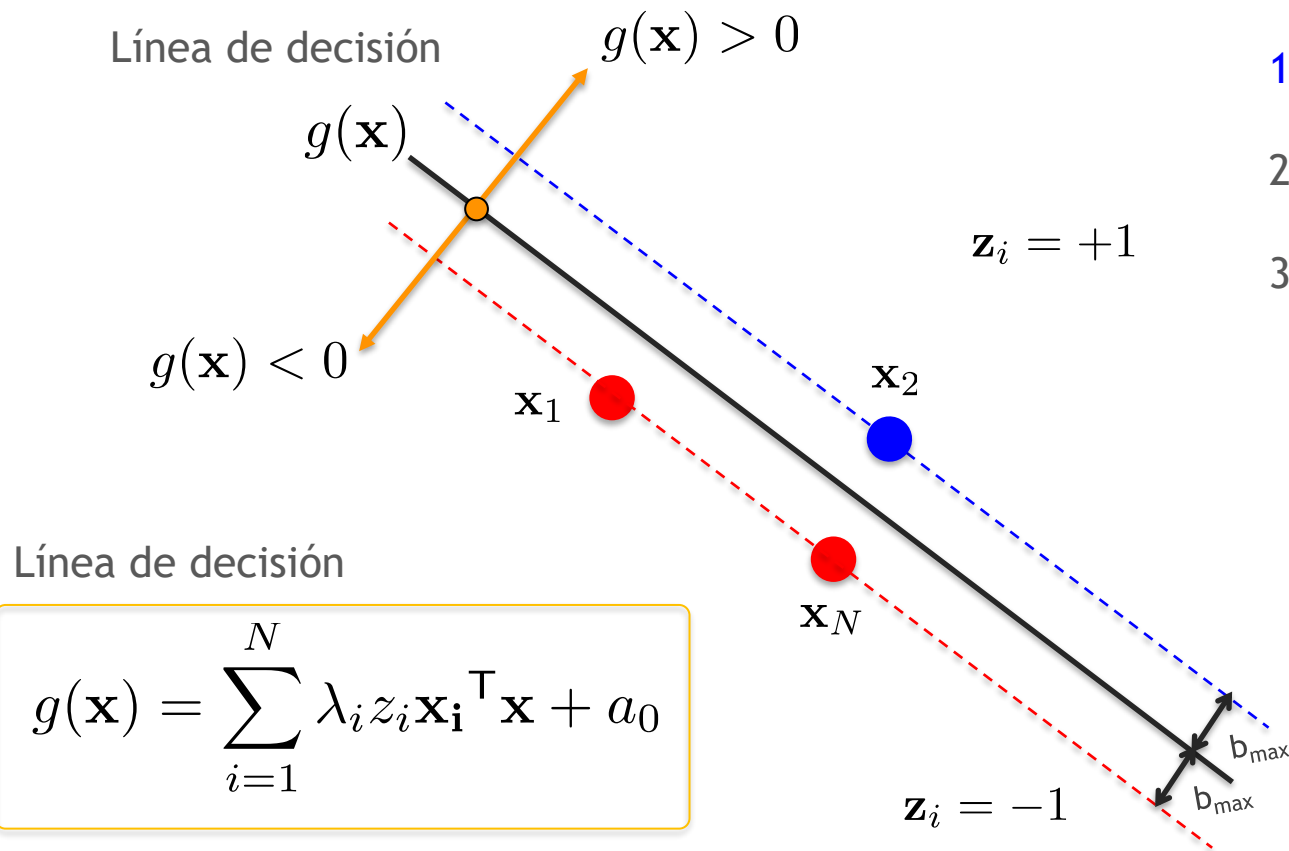
## SVM: dos clases



Ideas clave de SVM:

- 1)  $b_1 = b_2 = b$ .
- 2)  $b$  debe maximizarse.
- 3) Solución:  $g(\mathbf{x})$

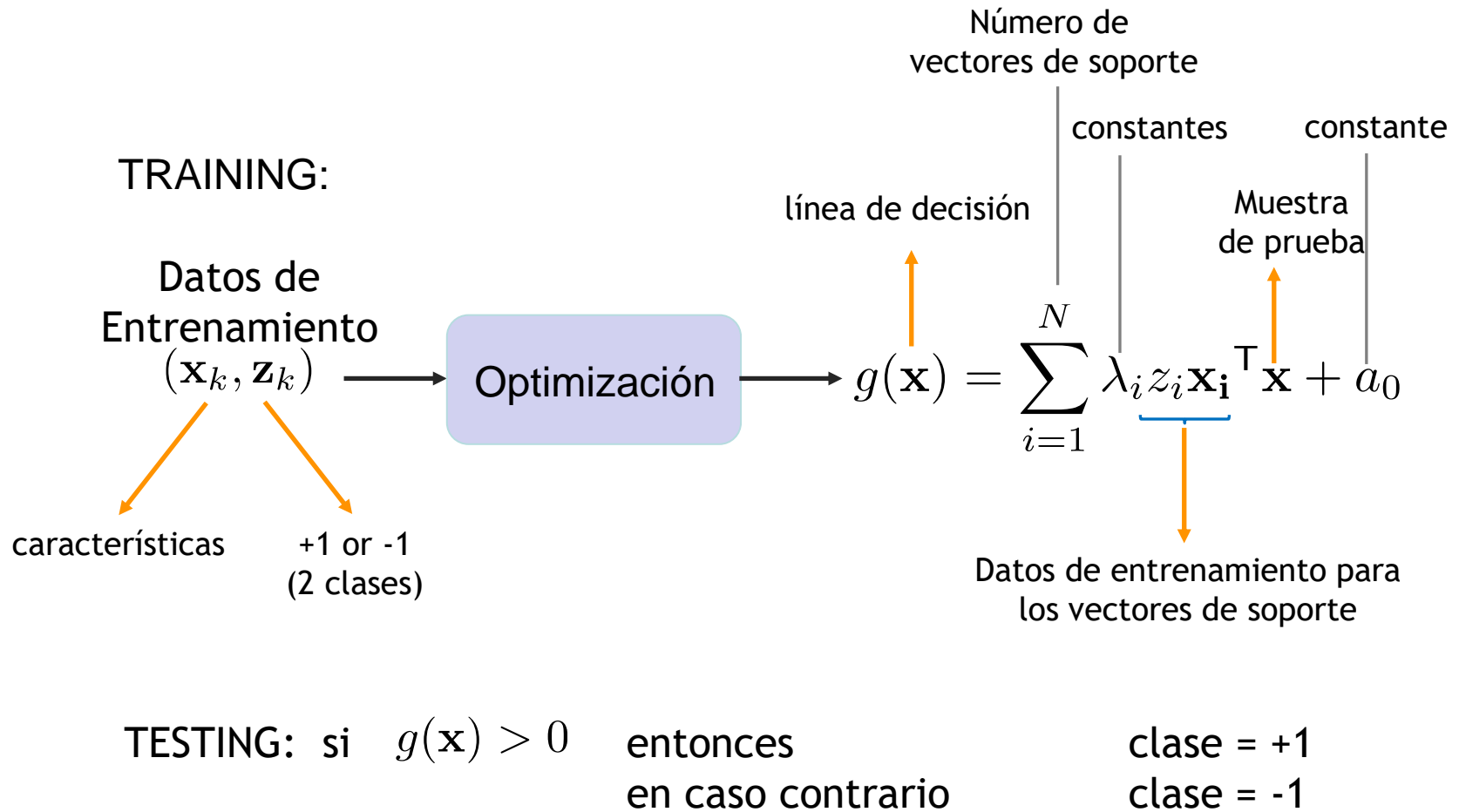
## SVM: dos clases



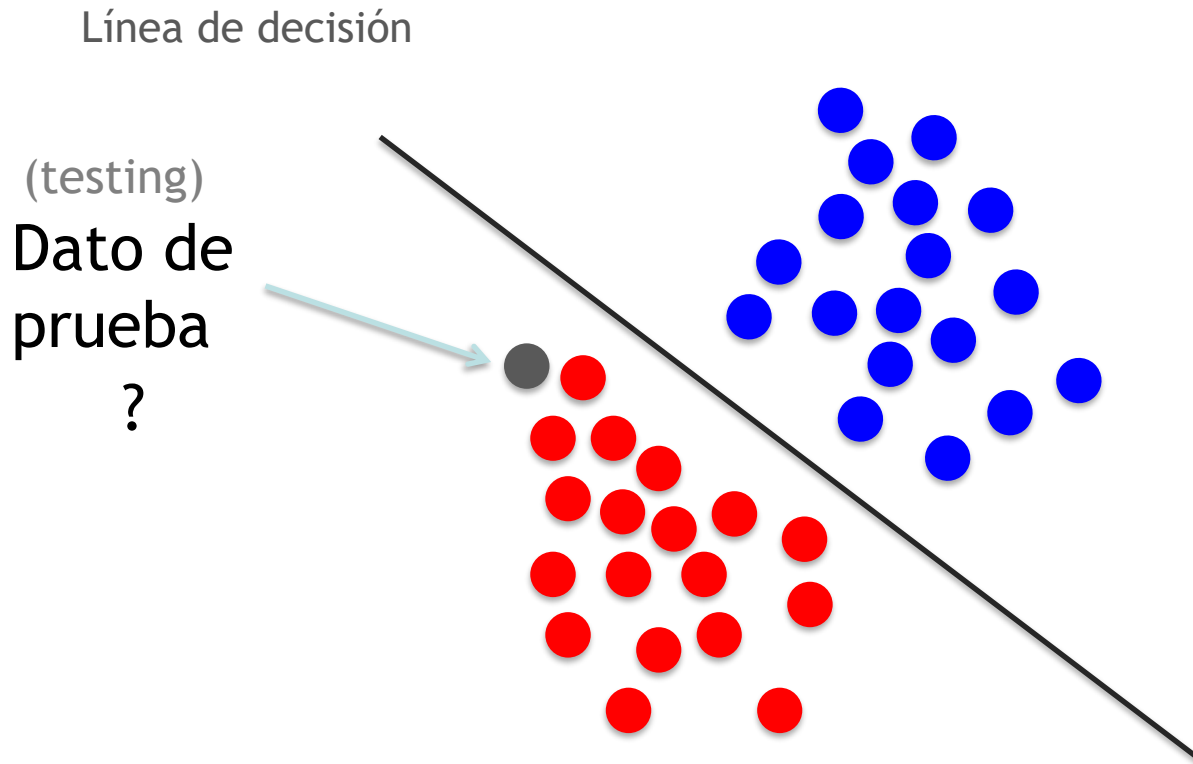
Ideas clave de SVM:

- 1)  $b_1 = b_2 = b$ .
- 2)  $b$  debe maximizarse.
- 3) Solución:  $g(\mathbf{x})$

# La solución de SVM

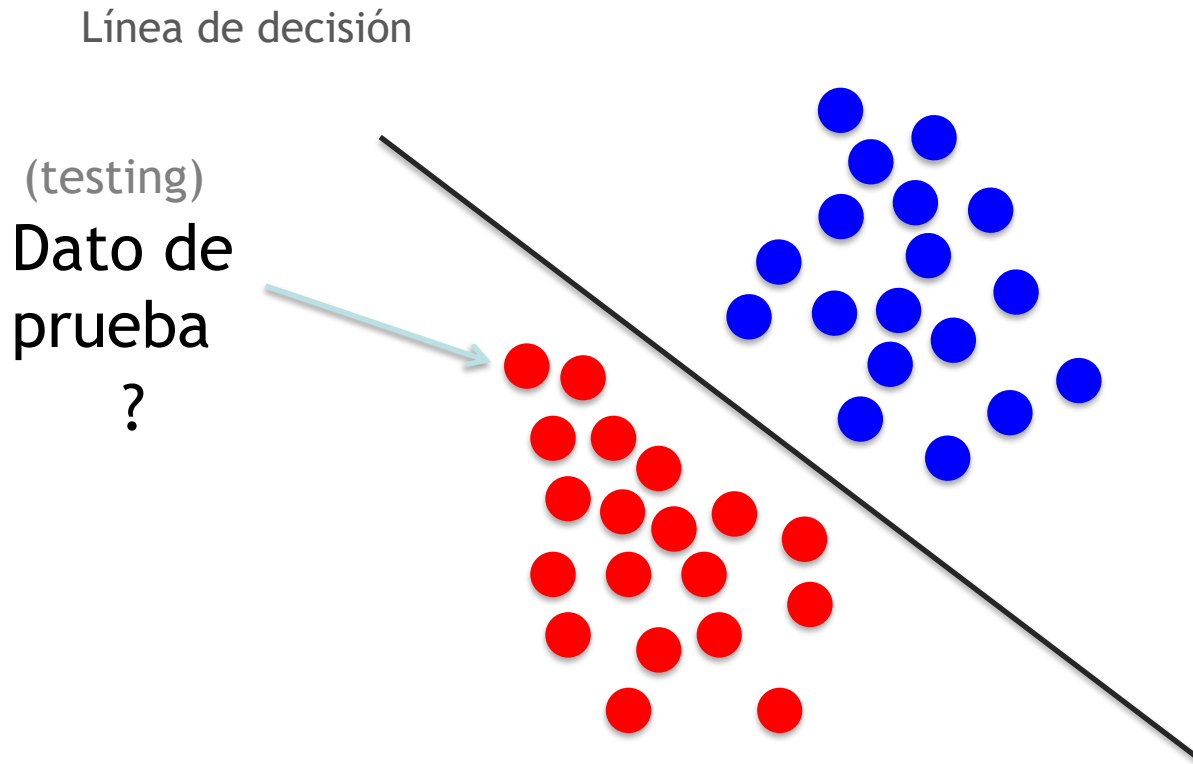


## SVM: dos clases



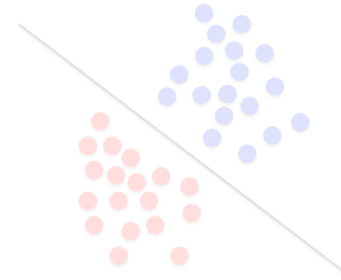


## SVM: dos clases

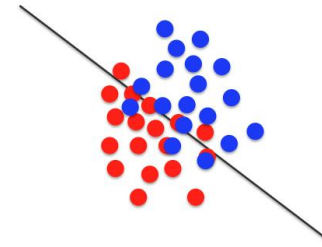


# SVM: Máquinas vectoriales de soporte

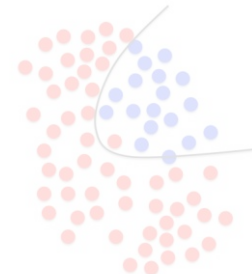
1) Lineal con separación perfecta



2) Lineal sin separación perfecta

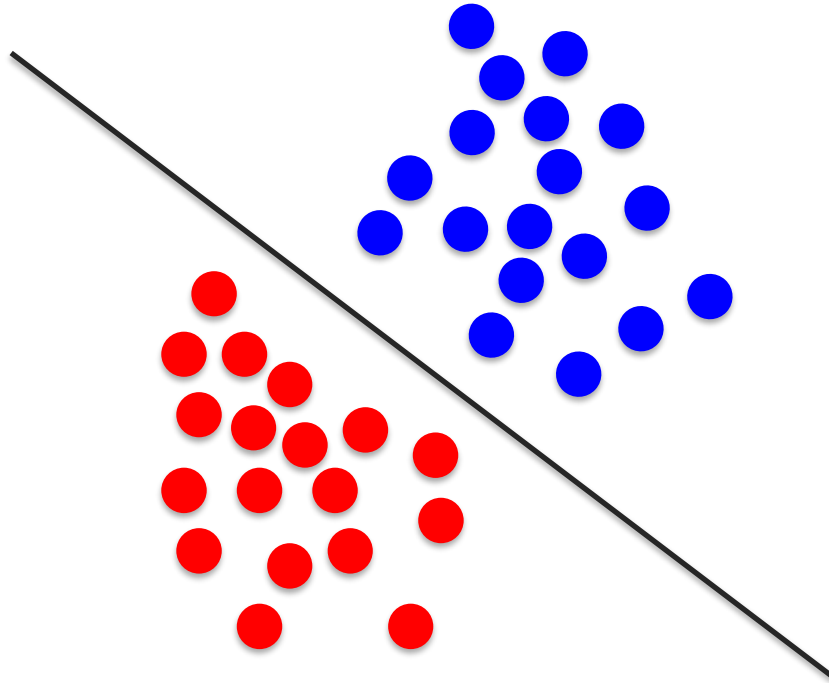


3) No lineal



## SVM: dos clases

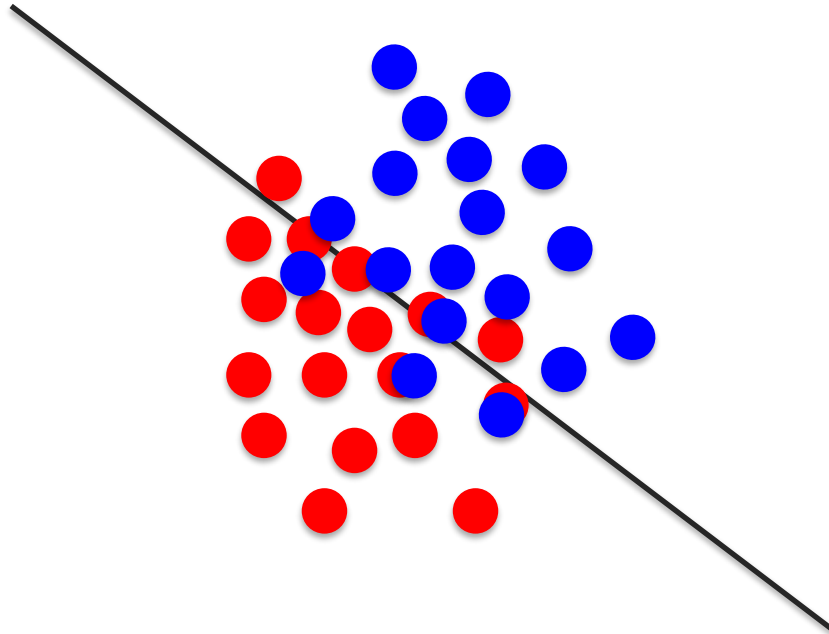
¿Cómo definir la línea de decisión cuando no hay una separación perfecta?



## SVM: dos clases

¿Cómo definir la línea de decisión cuando no hay una separación perfecta?

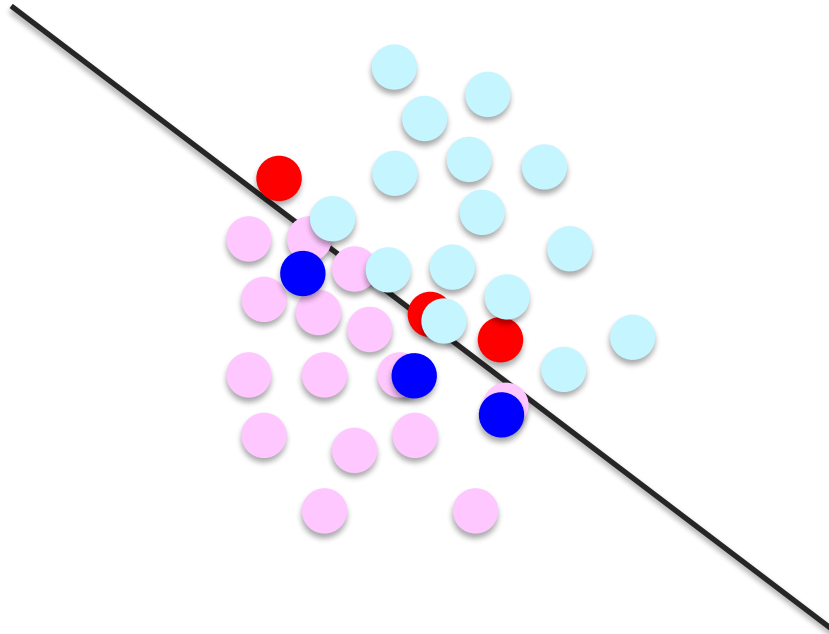
Consideramos sólo las muestras clasificadas erróneamente



## SVM: dos clases

¿Cómo definir la línea de decisión cuando no hay una separación perfecta?

Consideramos sólo las muestras clasificadas erróneamente

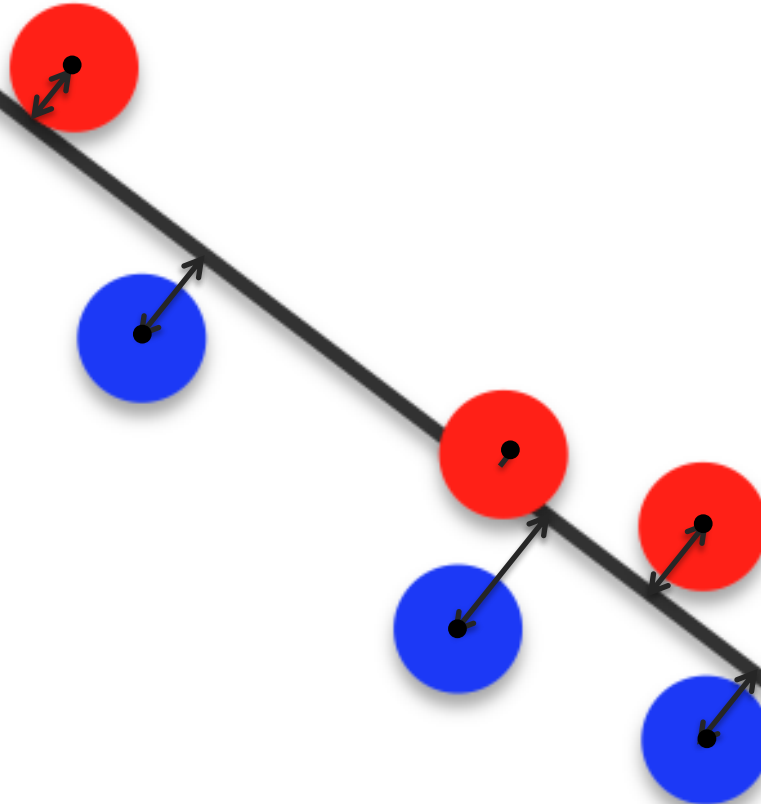


SVM

¿Cómo  
decisi  
separ

muestras  
ente

Vectores de soporte



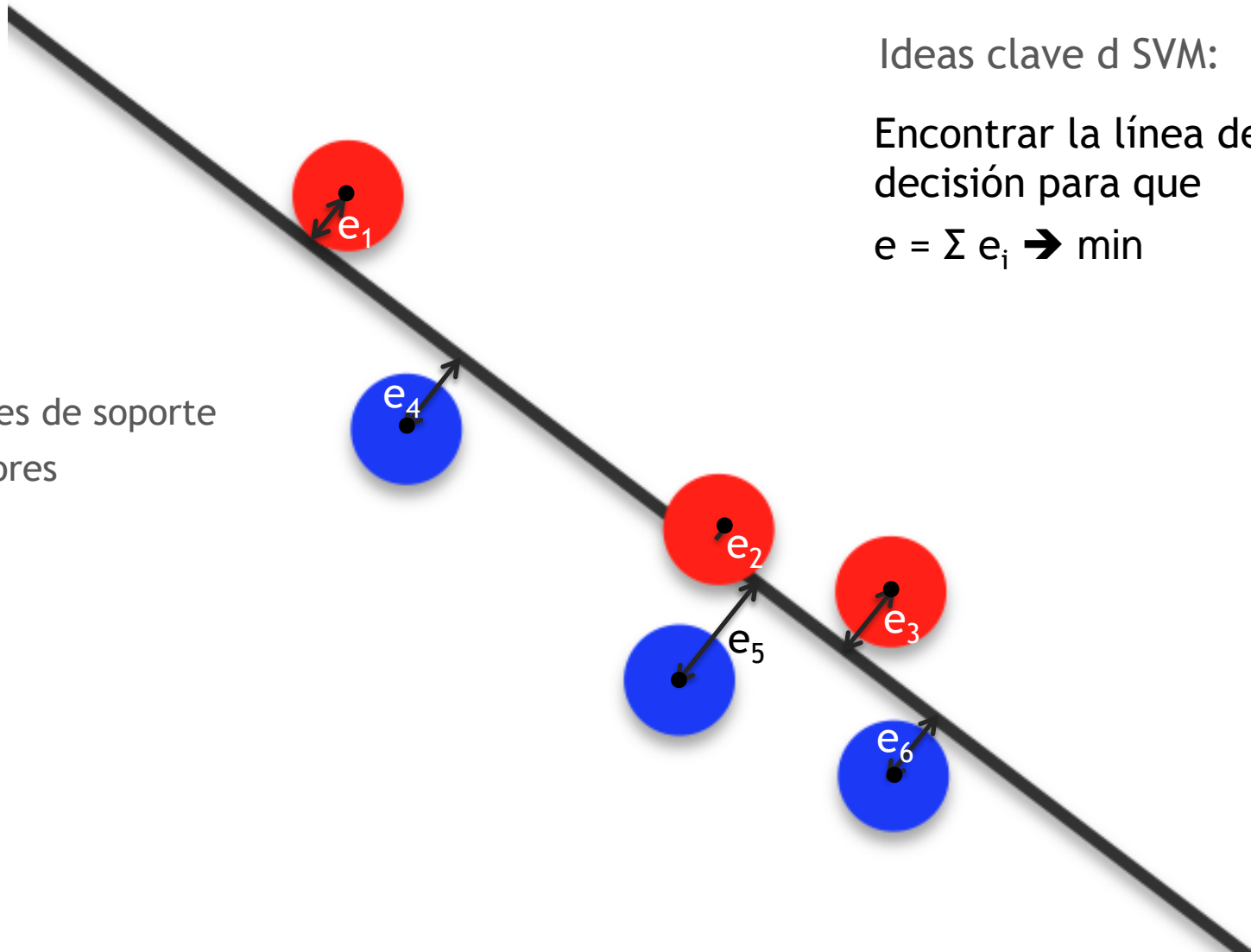
Ideas clave d SVM:

Encontrar la línea de  
decisión para que

$$e = \sum e_i \rightarrow \min$$

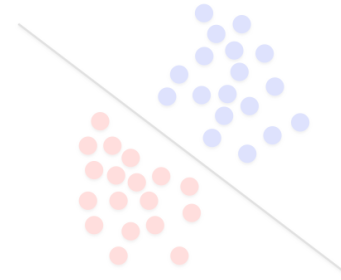
Vectores de soporte

$e_i$ : errores

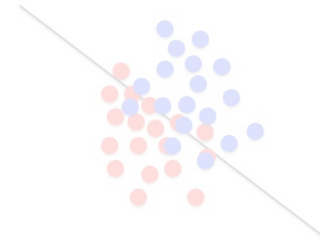


# SVM: Máquinas vectoriales de soporte

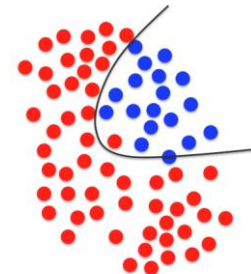
1) Lineal con separación perfecta



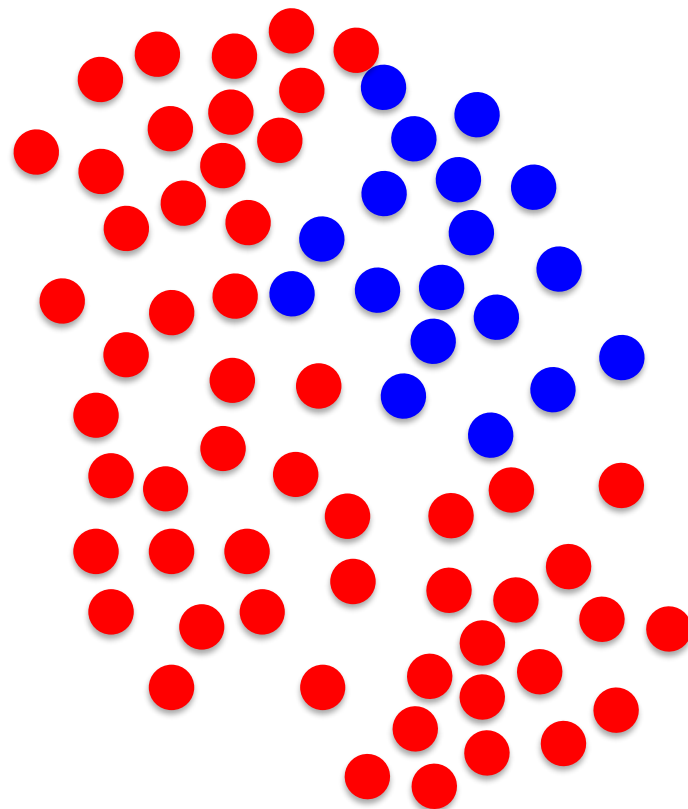
2) Lineal sin separación perfecta

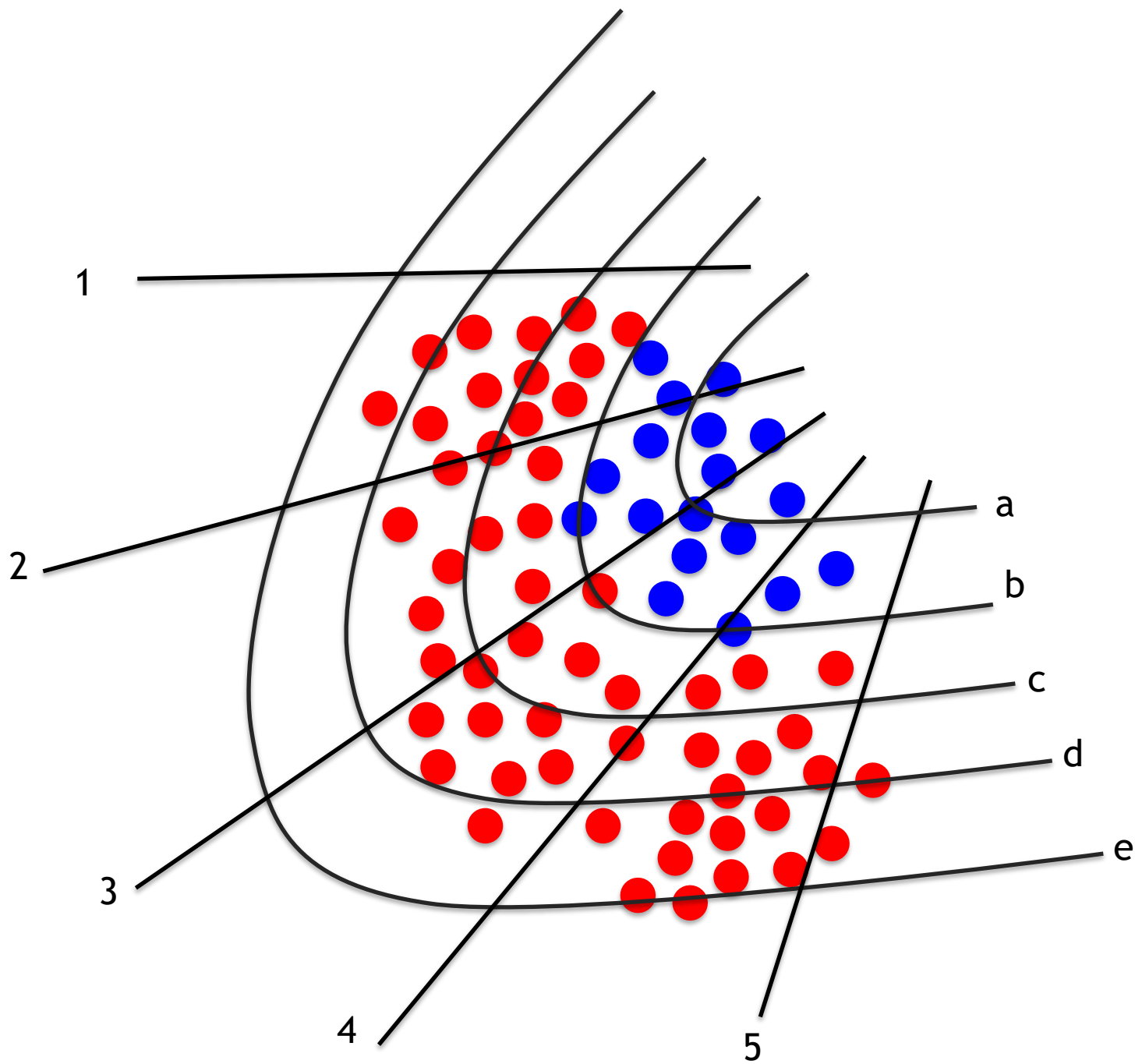


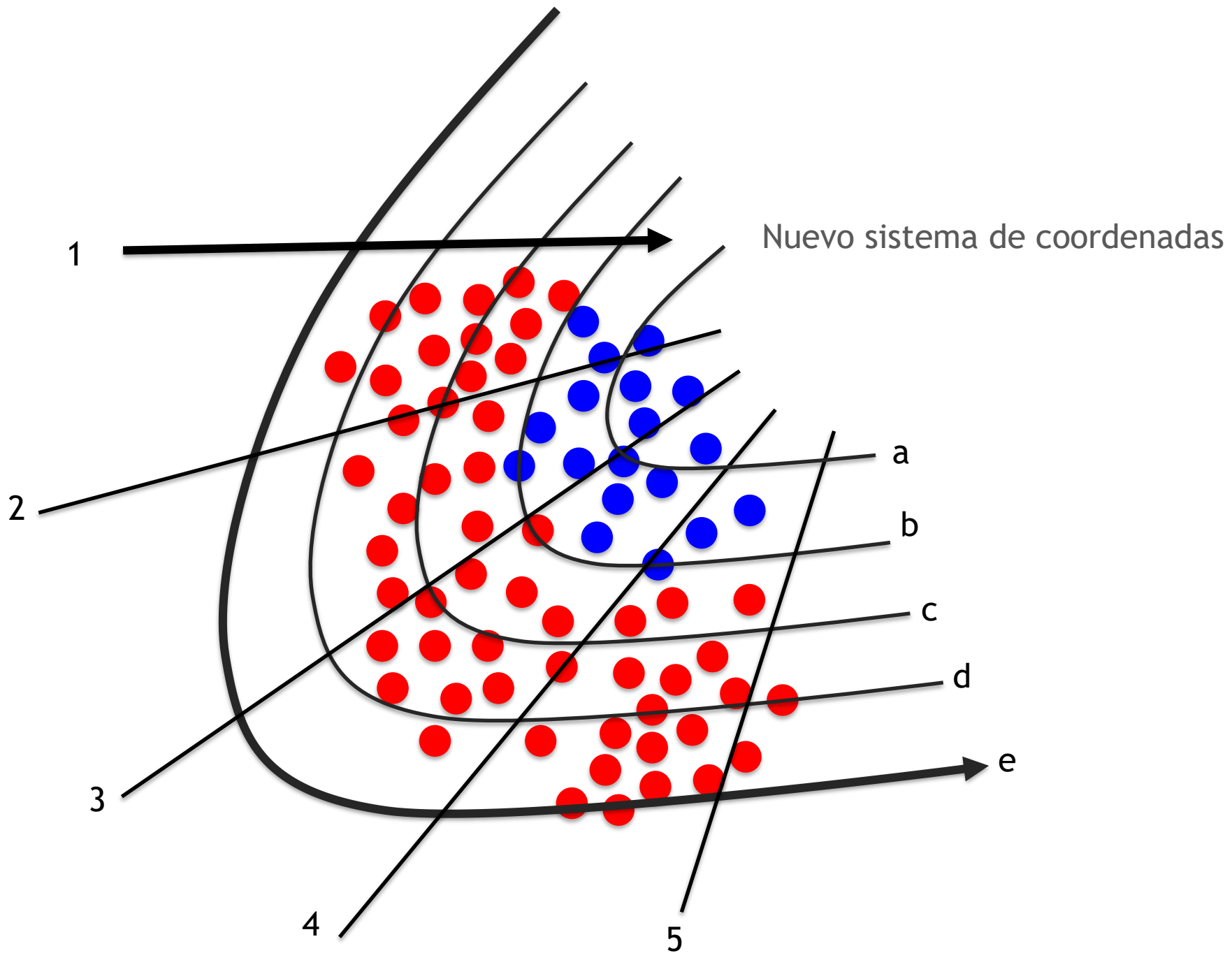
3) No lineal

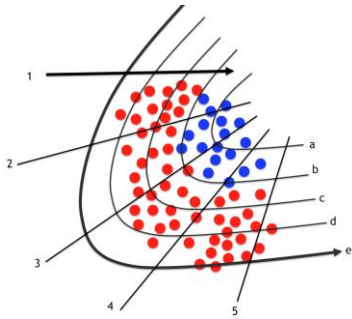




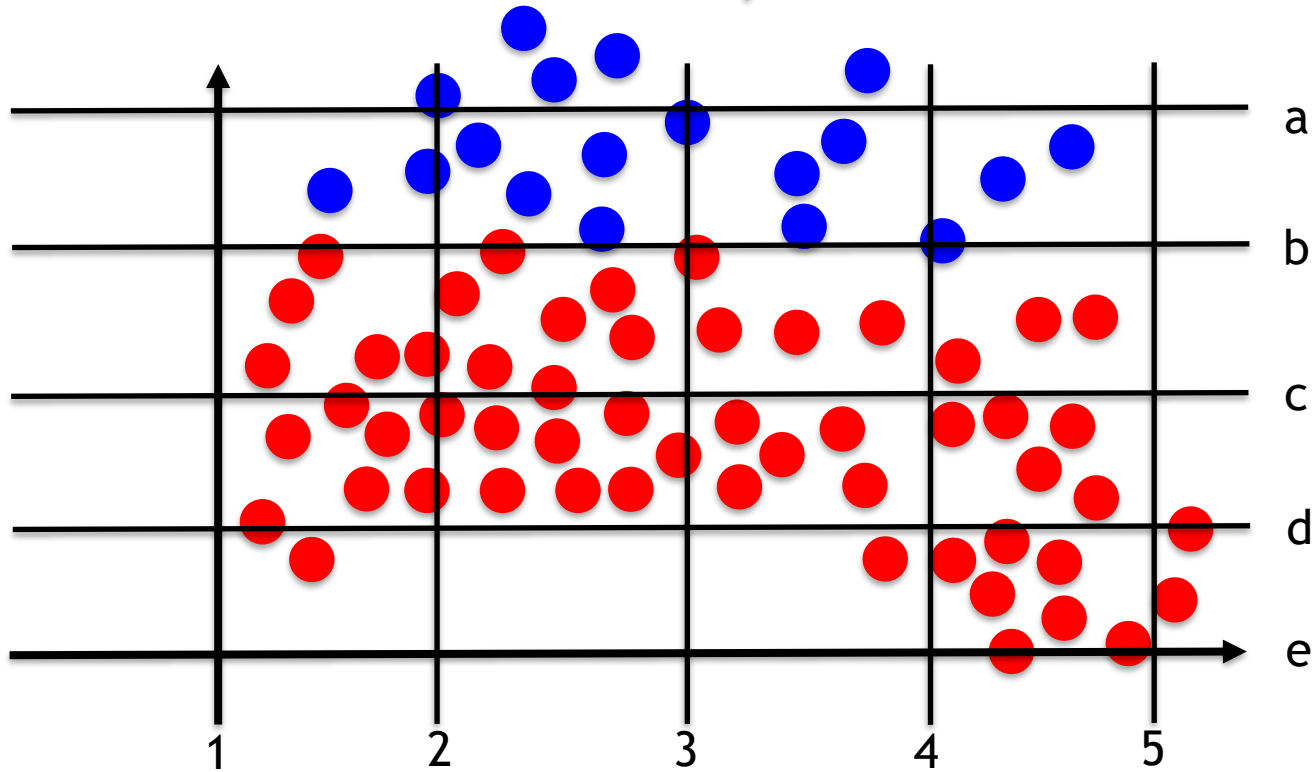


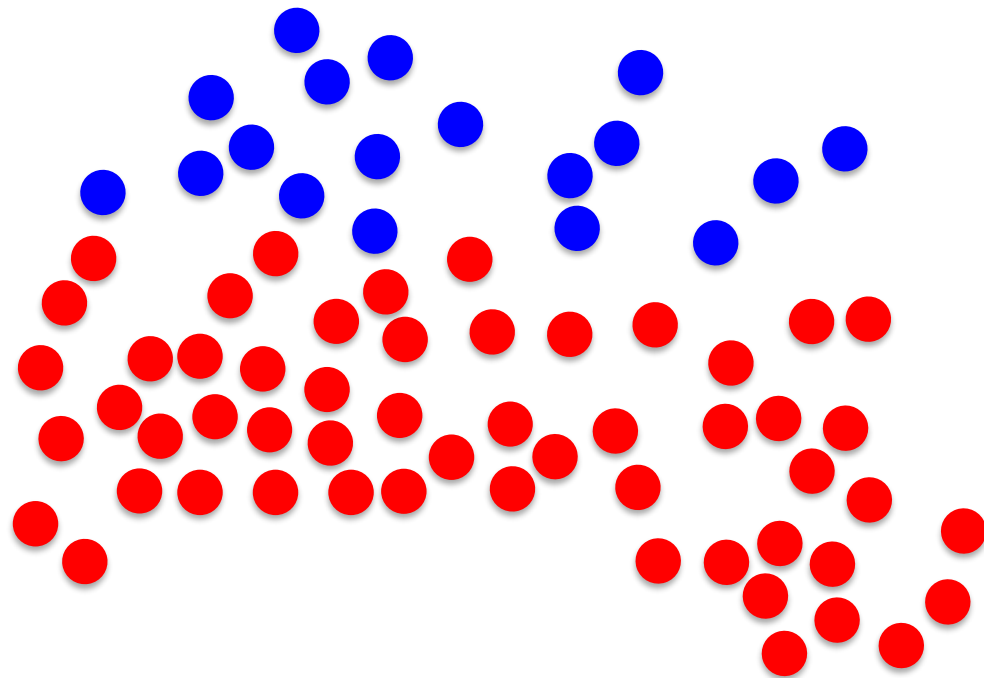




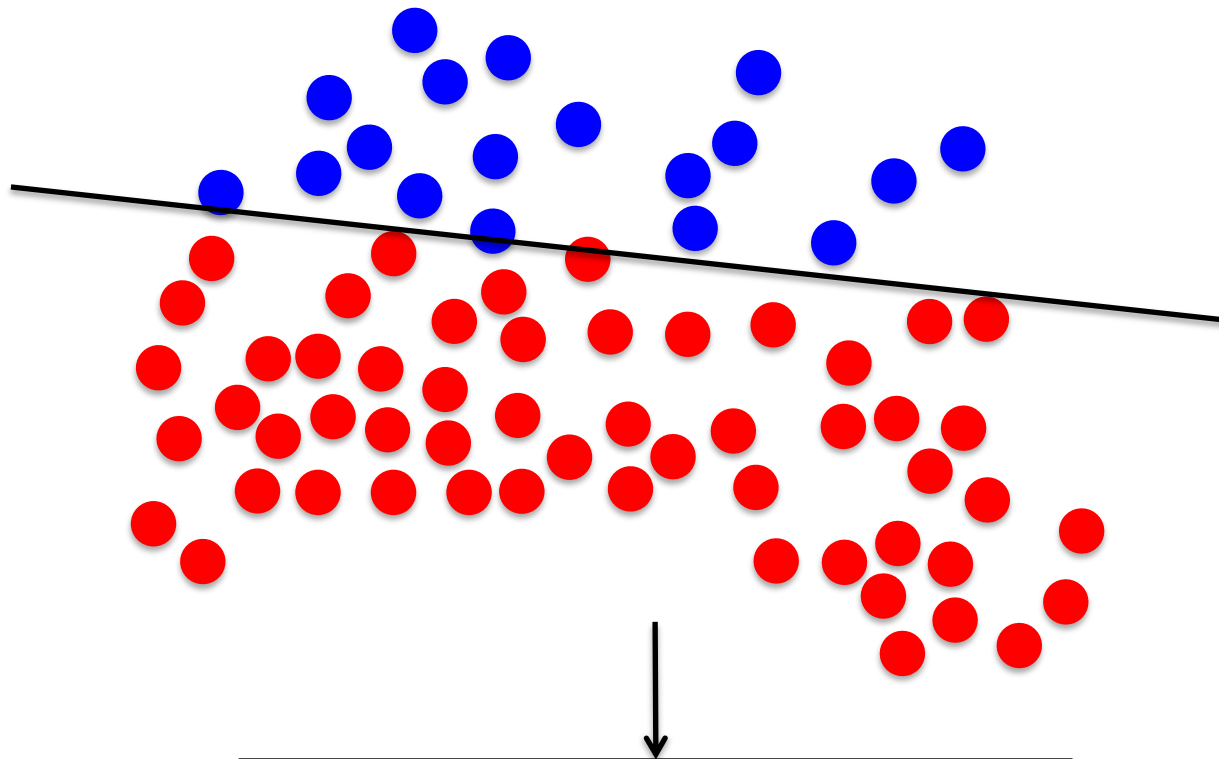


Transformación geométrica no lineal

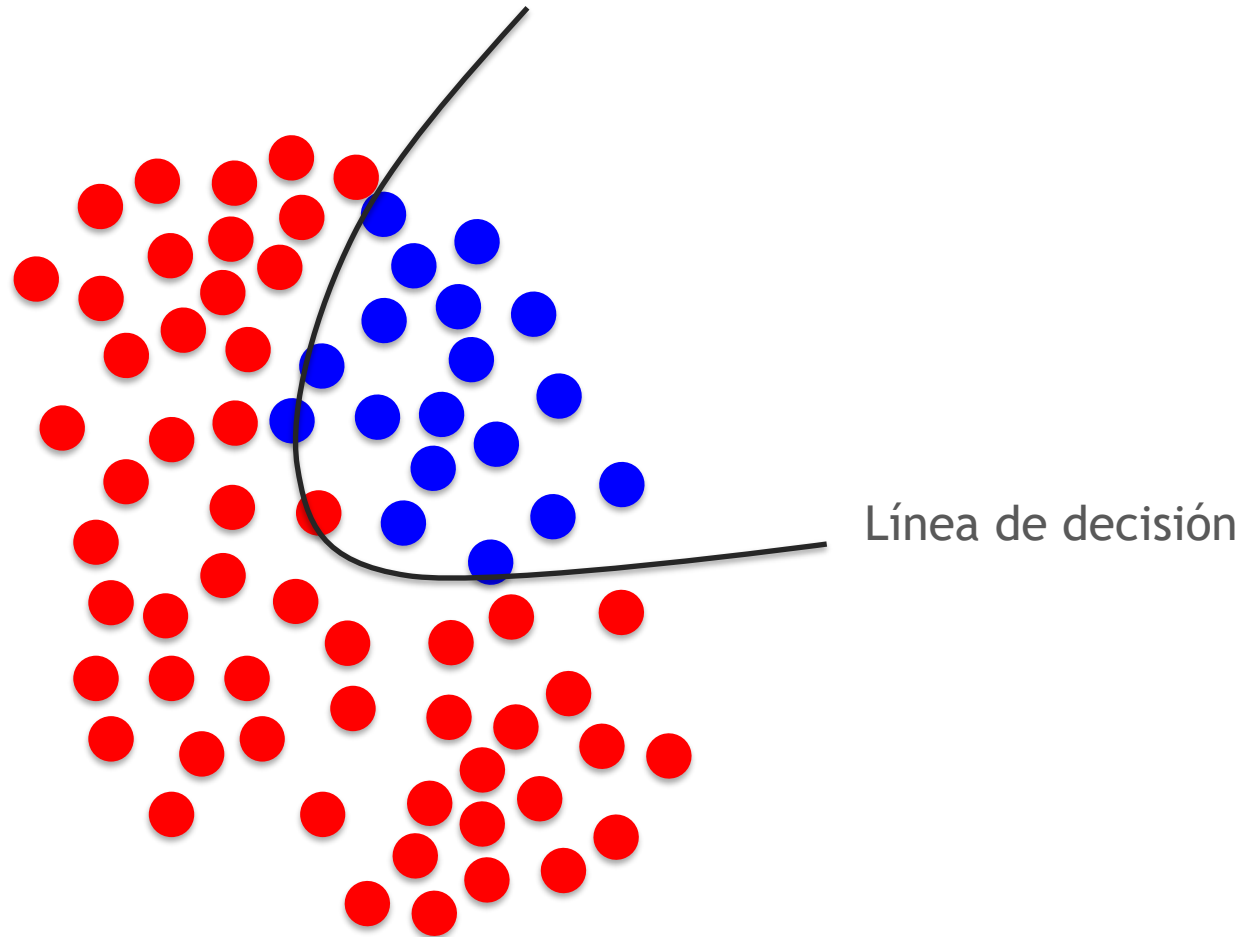




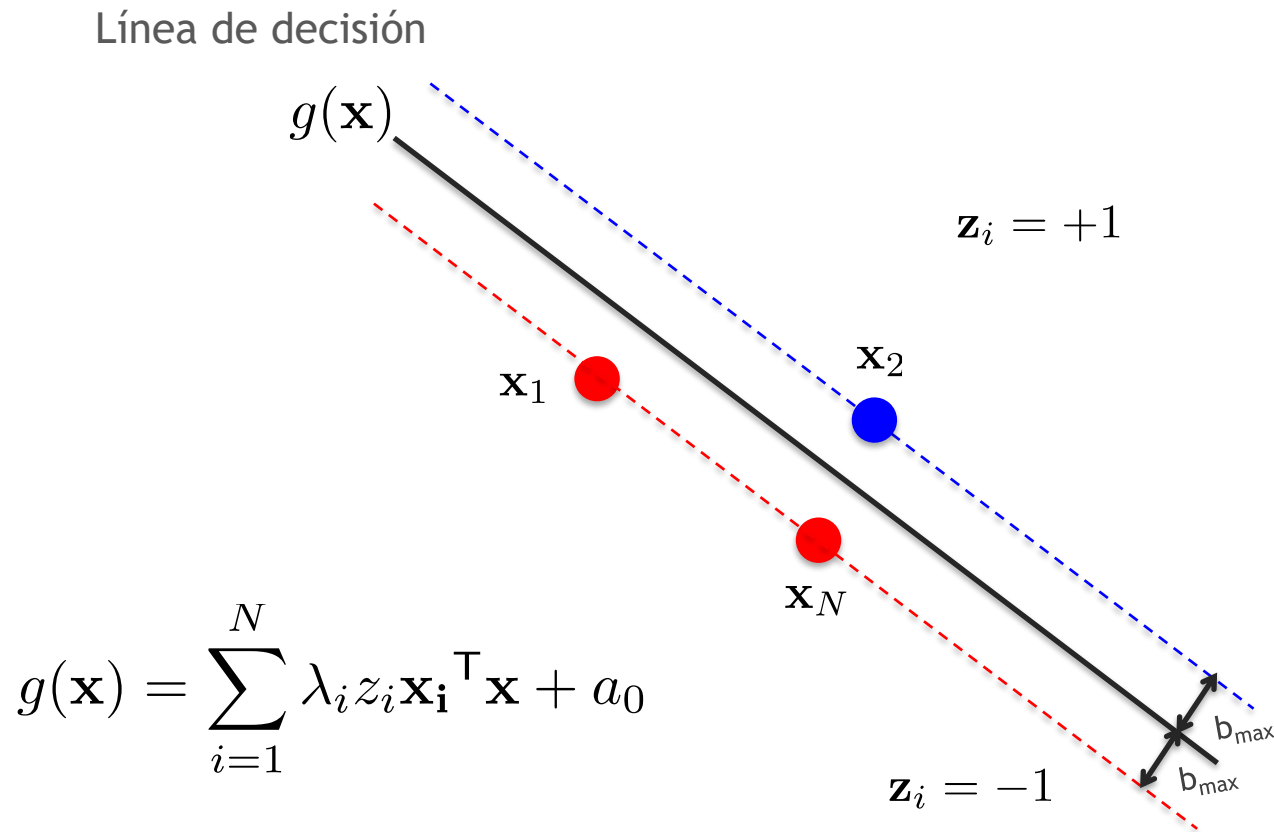
## SVM lineal en nuevo sistema de coordenadas



Transformación geométrica no lineal

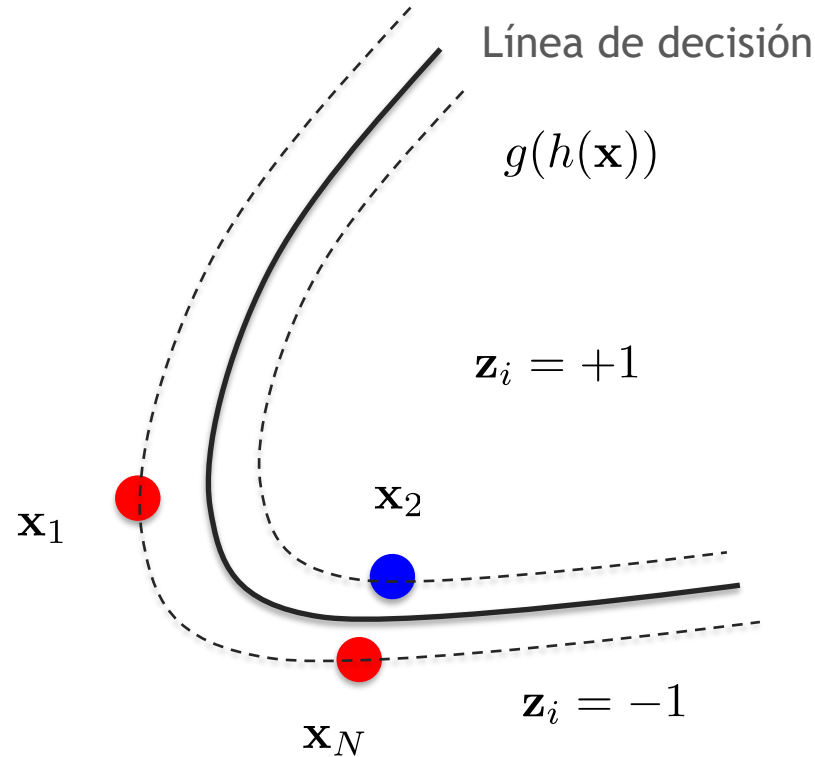


# SVM: Máquinas vectoriales de soporte





## SVM: Máquinas vectoriales de soporte



$$g(h(\mathbf{x})) = \sum_{i=1}^N \lambda_i z_i h(\mathbf{x}_i)^T h(\mathbf{x}) + a_0$$

$$g(h(\mathbf{x})) = \sum_{i=1}^N \lambda_i z_i \langle h(\mathbf{x}_i), h(\mathbf{x}) \rangle + a_0$$

No se necesita  $h(\mathbf{x})$ , solo es necesario el kernel  $\langle h(\mathbf{x}_i), h(\mathbf{x}) \rangle$

## SVM: Kernels

$$K(\mathbf{x}', \mathbf{x}) = \langle h(\mathbf{x}'), h(\mathbf{x}) \rangle =$$

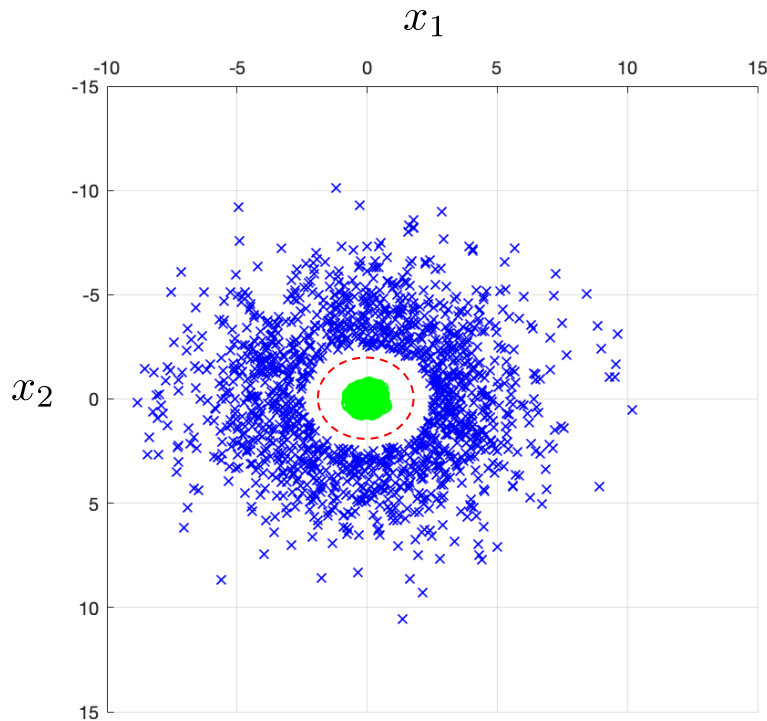
linear	$\langle \mathbf{x}', \mathbf{x} \rangle$
--------	---

polynomial	$(1 + \langle \mathbf{x}', \mathbf{x} \rangle)^n$
------------	---

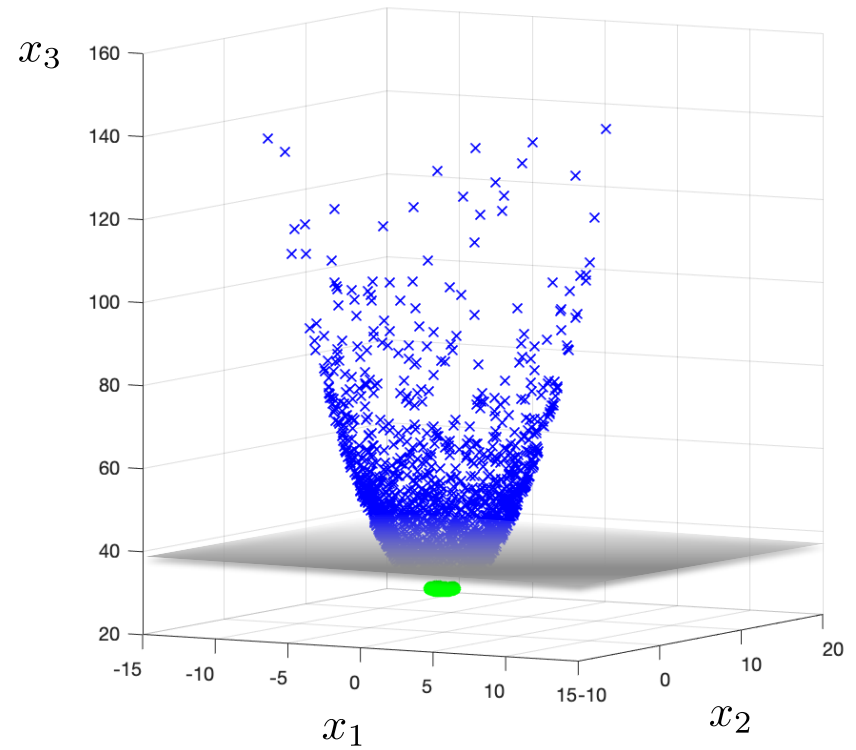
radial basis	$\exp(-\ \mathbf{x}' - \mathbf{x}\ ^2 / c)$
--------------	---

sigmoid	$\tanh(K_1 \langle \mathbf{x}', \mathbf{x} \rangle + K_2)$
---------	--

# SVM: El truco del Kernel



La separación lineal es imposible

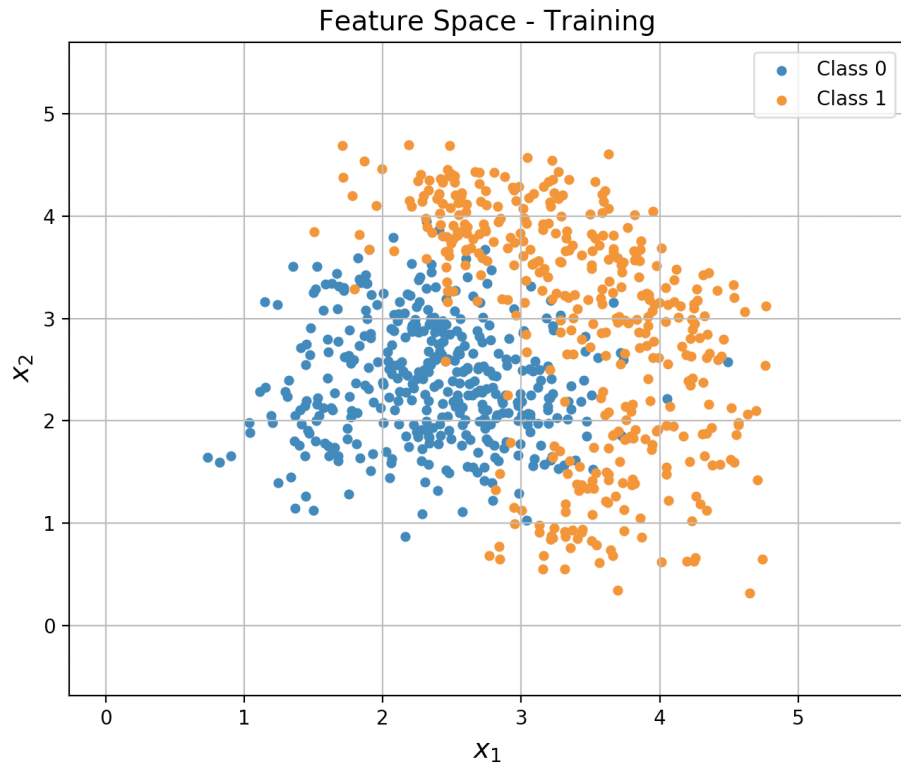


La separación lineal es perfecta

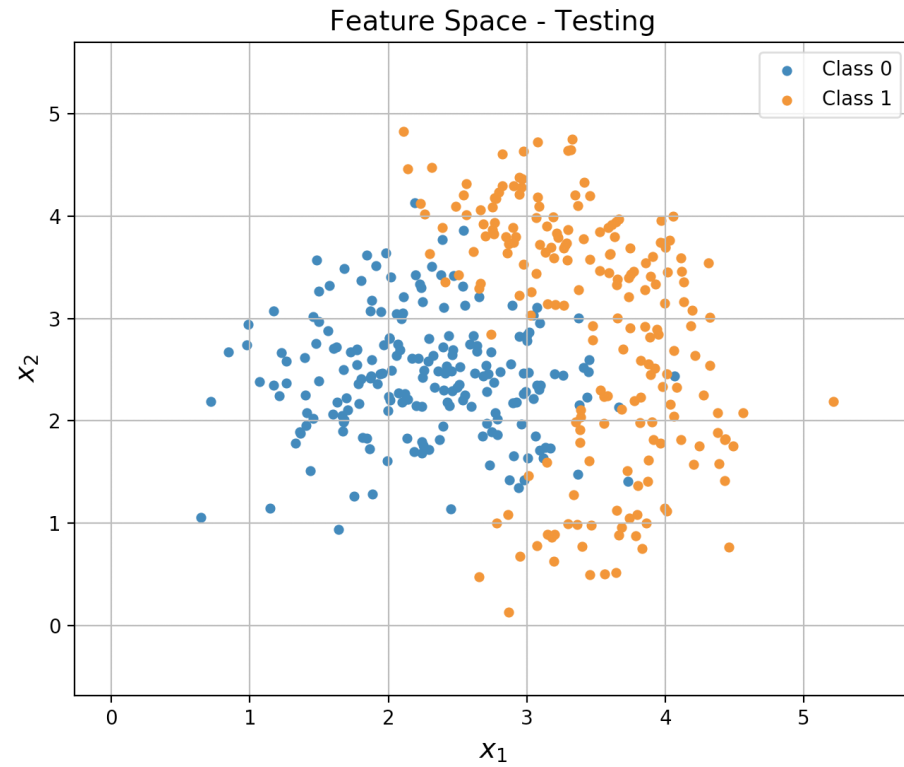
Ejemplos

# Ejemplo

$(X, d)$   
TRAINING



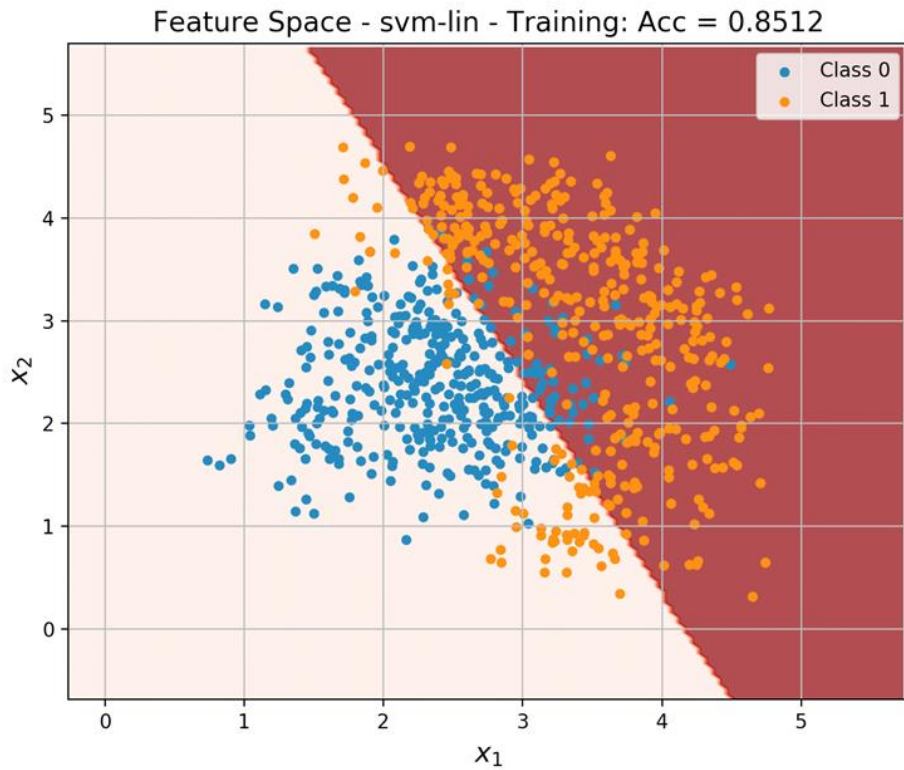
$(X_t, d_t)$   
TESTING



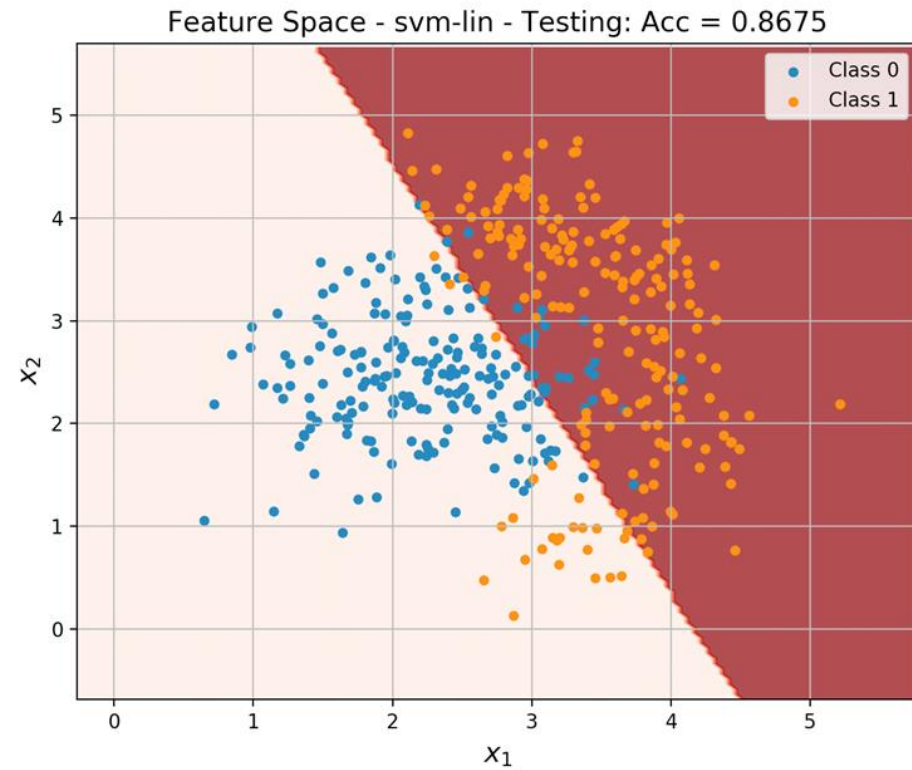
# Ejemplo

## SVM-LIN

$(X, d)$   
TRAINING



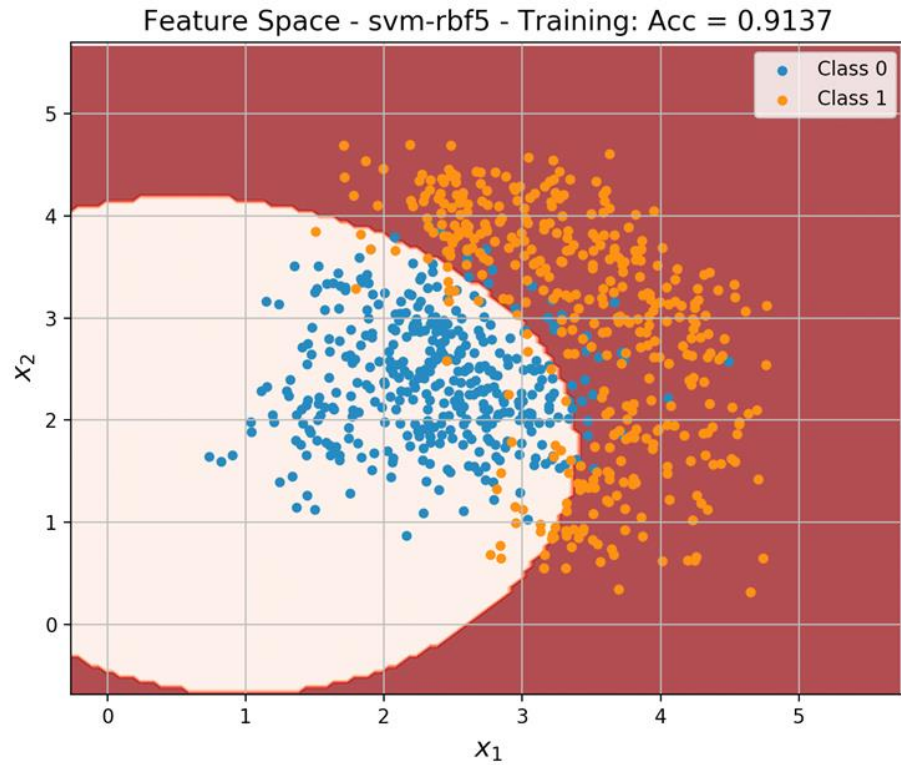
$(X_t, d_t)$   
TESTING



# Ejemplo

## SVM-RBF

$(X, d)$   
TRAINING



$(X_t, d_t)$   
TESTING

